Computer Science • 13 (1) 2012 http://dx.doi.org/10.7494/csci.2012.13.1.49

Ibrahim Elsayed
Peter Brezany

DATASPACE SUPPORT PLATFORM FOR E-SCIENCE

Abstract *This work intends to provide a data management solution based on the concepts of dataspaces for the large-scale and long-term management of scientific data. Our approach is to semantically enrich the existing relationship among primary and derived data items, and to preserve both relationships and data together within a dataspace to be reused by owners and others. To enable reuse, data must be well preserved. Preservation of scientific data can best be established if the full life cycle of data is addressed. This is challenged by the e-Science life cycle ontology, whose major goal is to trace the semantics about procedures in scientific experiments. We present a theoretical dataspace model for e-Science applications, its implementation within a dataspace support platform and an experimental evaluation on top of two real world application domains.*

Keywords scientific data management, dataspace, dataspace support platform, e-Science

1. Introduction

Dataspaces are not a data integration approach, rather they are a data co-existence approach [12]. The goal is to raise the abstraction level at which data is managed. Dataspaces are modeled as participants and relationships. Participants can be any data element and relationships should be able to model any interconnection among these participants. Dataspace support platforms (DSSPs) represent the collection of software programs and services that control the organization, storage and retrieval of data in a dataspace. The challenges of dataspaces discussed in [9] have influenced many research groups of the data management community. However, most effort was put on mainstream related dataspace research like indexing dataspaces [7] and pay-as-you-go data integration approaches for dataspace systems [5, 14] or on development of personal dataspace systems [6, 15].

In our previous work we addressed the dataspace-aspect regarding the creation and management of semantically rich relationships among dataspace participants. We introduced the e-Science life cycle ontology [9], which addresses the precise description of scientific experiments by taking advantage of the well-defined semantics of the Resource Description Framework (RDF) [19] and the expressive formal logic-based OWL language [20]. It is used to trace the semantics about procedures in e-Science applications. These procedures are modeled in five phases, which we name the *e-Science life cycle activities*. They classify at a high level of abstraction the activities a scientist is carrying out when performing a scientific experiment. Individuals of the ontology (expressed in RDF) represent descriptions of conducted experiments and thus aim at understanding (1) what for a specific experiment was applied, (2) which data resources were accessed, (3) what transformations on these data resources were applied, (4) what analysis were performed, and finally (5) what results where achieved. The ontology also provides a publication concept allowing acting scientists to set publication modes to applied experiments. Thus, access to specific experiments can be limited to specific members or groups of the scientific community.

In [8, 9] we introduced the scientific dataspace, which aims at providing associated mechanisms for managing semantically rich relationships among scientific data sources (primary data) and its corresponding findings (derived data), which result from a set of activities defining concrete preprocessing and analysis methods (background data) that were applied on a dataset. Furthermore, to keep track of scientific experiments that are being conducted by members of a scientific community and to link these experiments with user information, i.e. institutional affiliation, email address, working field, etc. of the scientist who conducted the experiment.

In this paper, we present a theoretical dataspace model for e-Science applications in Section 2 and its implementation (Section 4) within the architecture (Section 3) of the dataspace support platform. We also present an experimental evaluation on top of two real word e-Science applications in Section 4 which outlines the performance overhead with the dataspace support platform.

Table 1
Dimensions of the Scientific Dataspace

Activity Name	Definition
Specify Goals	$GS = \{I_{GS1}, I_{GS2}, ..., I_{GSn}\}$
Prepare Data	$DP = \{I_{DP1}, I_{DP2}, ..., I_{DPn}\}$
Select Appropriate Tasks	$TS = \{I_{TS1}, I_{TS2}, ..., I_{TSn}\}$
Run Tasks	$TE = \{I_{TE1}, I_{TE2}, ..., I_{TEn}\}$
Process & Publish Results	$RP = \{I_{RP1}, I_{RP2}, ..., I_{RPn}\}$

2. The Scientific Dataspace Model

Scientific experiments described by the e-Science life cycle ontology are referred to as *Life Cycle Resources* (LCR). They are organized as points of a 5-dimensional space, where dimensions represent the five e-Science life cycle activities. Coordinates of a dimension represent the individuals of the corresponding activity. A dimension of the space can be regarded as a 1-dimensional space. Table 1 lists the names of the e-Science life cycle activities defined in the e-Science life cycle ontology and their definitions in the scientific dataspace model.

Coordinates have a name, a set of properties and a set of individuals that are interconnected via those properties. Coordinates are defined as $I_{Xn} = \{name, P, E\}$, with $X \in \{GS, DP, TS, TE, RP\}$, where P refers to the set of properties defined for the coordinate and E refers to the set of elements that are connected to the coordinate. Elements are instances of classes of the e-Science life cycle ontology. Thus, a coordinate represents a graph, with I_{Xn} as root element and $e \in E$ as elements of the graph interconnected via properties $p \in P$.

A point in the multidimensional space connects coordinates (i.e. individuals of the five e-Science life cycle activities) that are participating within a LCR. A LCR can therefore be defined as a vector:

$$v_i \left\{ I_{GSj}\ I_{DPk}\ I_{TSl}\ I_{TEm}\ I_{RPn} \right\}$$

with $j, k, l, m, n \in \mathbb{N}$ representing the index of the five participating coordinates.

Figure 1 (a) illustrates a 3-dimensional space. The point depicted as $LCR(I_{GS_4}|I_{DP_4}|I_{TS_4})$ in Figure 1 (a) represents a *LCR* with three individuals of the e-Science life cycle activities *"Specify Goals", "Prepare Data"* , and *"Select Appropriate Tasks"*. Information about actions taken within an activity is saved as an RDF graph with the individual as root element.

Spaces with more than three dimensions are hard to project on 3-dimensional images. One way is to split the n-dimensional cube into multiple 2-dimensional cubes and to position them accordingly. In Figure 1 (b) we try to visualize the 5-dimensional life cycle dataspace. We should keep in mind that the five 2-dimensional points depicted in Figure 1 (b), actually form together a single point in the 5-dimensional space, therefore represent a single LCR.

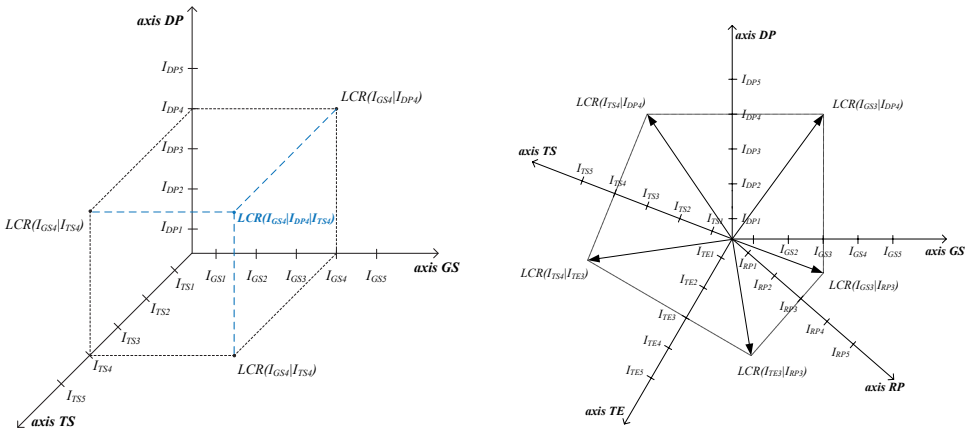

Figure 1. 3- and 5-dimensional Scientific Dataspace

Mathematically the 5-dimensional space is defined as a $n \times 5$ matrix. Elements of the matrix are defined as a_{ij} with $i = 1, ..., m$ (the index of coordinates corresponding to a LCR) and $j = 1, ..., 5$ (the index of the e-Science life cycle activity). Each element represents an individual of an e-Science life cycle activity. Columns represent the dimensions of the multidimensional space and rows represent the points (LCRs) of the space. Points in the space can also be combined of coordinates having different indexes, e.g. a LCR using individual p of the dimension *"Specify Goals"* can contain instances with a different index on the other dimensions. This is important when a researcher reuses an available individual within another iteration of the e-Science life cycle. For example, when applying the same analysis method on a different dataset (that is a different instance of the *"Prepare Data"* activity). Let's assume that the three instances I_{GS3}, I_{TE3} and I_{RP3} are being reused within a new e-Science life cycle experiment. This indicates that the acting researcher has applied a new analysis method on a new prepared dataset, but reused the same instance of the dimension *"Specify Goals"*, *"Run Tasks"* and *"Process & Publish Results"*, thus working on the same study, executing the analytical methods on the same machine and publishing results using the same publication modes as in his previously conducted experiments. The corresponding LCR is illustrated in Figure 1 (b) and denoted as

$$LCR = v_i \left\{ I_{GS3}\ I_{DP4}\ I_{TS4}\ I_{TE3}\ I_{RP3} \right\}\ with\ i \in \mathbb{N},\ i < n$$

n is the number of rows in the corresponding matrix and is equal to the amount of LCRs. This measure indicates the state of the scientific dataspace.

If we assume that the LCR depicted in Figure 1 (b), has the highest available index in the dataspace, therefore represent its actual state, we can organize the

5-dimensional scientific dataspace in the following 4×5 matrix:

$$A_{LCR} = (I_{45}) \begin{pmatrix} I_{GS1} & I_{DP1} & I_{TS1} & I_{TE1} & I_{RP1} \\ I_{GS2} & I_{DP2} & I_{TS2} & I_{TE2} & I_{RP2} \\ I_{GS3} & I_{DP3} & I_{TS3} & I_{TE3} & I_{RP3} \\ 0 & I_{DP4} & I_{TS4} & 0 & 0 \end{pmatrix}$$

The positions I_{14}, I_{44} and I_{45} contain the value 0, which indicates that there still doesn't exist an individual of the dimensions $1, 4$ and 5 having index 4. With an increasing number of life cycle experiments the dataspace and therefore the matrix A_{LCR} increases. However, the index is separately updated for each dimension and only when a new individual of the corresponding life cycle activity is created.

3. Platform Architecture

In this Section, we provide a summary of the architecture of a scientific dataspace support platform. The main entities of the architecture are the *Life Cycle Composer* – for the creation of LCRs, the *RDF Store* – for storing those resources, the *Dataspace Indexer* – for their subscription, the *Search&Query Processor* allowing scientists to find those LCRs, and the *Dataspace Browser* for the exploration of the dataspace. These, with each other cooperating software programs represent the environment in which the scientific dataspace is able to grow and evolve into a remarkable space of well preserved scientific data. They also provide the organization and retrieval of scientific data within the dataspace. A holistic view of the architecture is given in Figure 2.

Search&Query Processor. Due to the fact that dataspace participants as well as their relationships are precisely described by the individuals of the e-Science life cycle ontology, therefore organized as RDF resources, we built the *Search&Query Processor* on top of the SPARQL query language [16], which has been accepted as a W3C recommendation for querying RDF resources. The *Search&Query Processor* consists of a *Query Interpreter* and a *Query Translator*. The query interpreter receives a request, which can be expressed either as a SPARQL Query or as a keyword based search. The request is forwarded to the *Query Translator*, who generates a SPARQL query (if not yet already expressed in SPARQL) out of the keywords. This SPARQL query is then submitted to the RDF store.

Searching and querying a dataspace in general is not like querying a database. In a dataspace we need to drift away from the one-shot query to query-by-navigation. Users will have to pose several queries, which results in an *Information Gathering Task* (IGT). IGT was introduced by Halevy et al. in [2] as one of the major principles of a dataspace system. This task is implemented as a multi-level process of submitting different types of queries. In the first level the RDF store, which organizes the individuals of the e-Science life cycle ontology, is queried while in following levels dataspace participants themselves are queried.

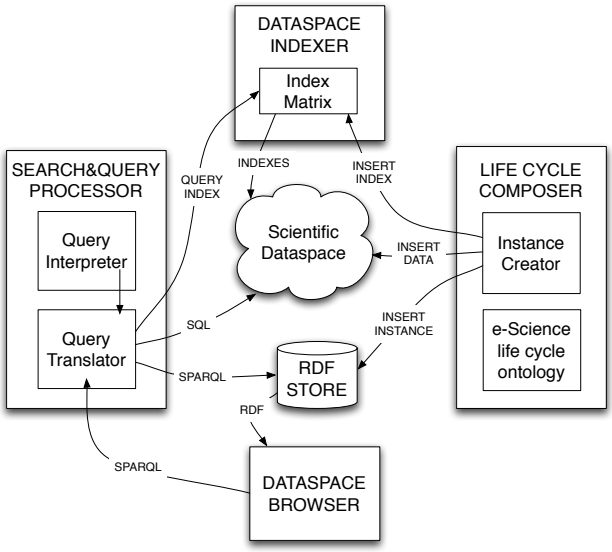

Figure 2. Holistic view of the system architecture [10]

The information that a scientist gathers in the first level represents the semantics about applied scientific experiments, like what were the research goals, what dataset was used, what analytical methods, etc. It will lead the scientist to those LCRs he might be interested in and to those that are interconnected to them. In the second level the data items that are used within previously identified LCRs can be retrieved. Such datasets are for example the input dataset used, or the dataset derived from selected scientific experiments. In order to apply this kind of deeper searching and querying more sophisticated queries are submitted to the scientific dataspace, in particular to the corresponding DBMS that participates in the dataspace. Depending on the data source it can be again SPARQL or SQL, XQuery, or any other query language that is supported by the underlying data source.

Scientific Dataspace and RDF Store. The scientific dataspace is modeled as set of dataspace participants and relationships describing their interconnections. A participant of the scientific dataspace is a dataset that either represents input data to a scientific experiment, or the analytical method being used within that experiment, or it is a dataset that has emerged during execution of an experiment. We therefore classify three types of participants: (a) primary data participants – the input dataset, (b) background data participants, i.e. an analytical method (web service, MATLAB script, etc.), and (c) derived data participants – emerged datasets.

Participants are described by its meta data. Meta data is organized by the OWL class *MetaData*, which is a generic class for describing individuals (e.g. participants) according to user-defined attributes. An instance of the

class *MetaData* typically has the form of a triple `<instanceID, attribute, value>`. For example, a short textual description of a participant could look like `<'participant073', 'description', 'breath gas analysis measurements taken from 20 probands at sleep laboratory Innsbruck'>`. With this data description concept, scientists can describe participants and also other individuals of the ontology according to their needs.

Relationships among the above described types of dataspace participants are semantically rich described by individual and property assertion axioms defined in a LCR. These axioms, consolidated within a LCR, describe on a semantically high level a scientific experiment. LCRs are expressed in RDF and persistently stored and are managed by the RDF store.

Relationships within the scientific dataspace model show how datasets (primary, background, or derived participants) were used in scientific experiments. Whenever a dataspace participant is retrieved by some kind of supported search&query mechanisms, the requesting user will automatically receive additional information about (1) which experiments (LCRs) the participant is involved in, (2) what the purposes of these corresponding experiments were, (3) which other participants are also involved in those experiments, therefore are interconnected to the retrieved participant, and (4) who the creator of the participant is, which research group he corresponds to and how to contact him.

This information is described by classes and properties of the e-Science life cycle ontology and it represents the semantically rich relationship among participants of the scientific dataspace.

Once a new e-Science life cycle experiment is composed via the *Life Cycle Composer* by a scientist, a new LCR is created, indexed and added to the RDF store. This process in described in the following subsections.

e-Science Life Cycle Composer. The e-Science life cycle composer enables a scientist to describe scientific experiments. It guides the user through the five e-Science life cycle activities, creates new individuals, and attaches them to a new LCR. It communicates with the *Dataspace Indexer*, which indexes new individuals.

The e-Science life cycle composer can be seen as the feeder interface to the scientific dataspace. It is the appropriate and easy to use way to enter semantically rich information about how the participating data items in the dataspace are related and interconnected together. A strong requirement here is to provide a simple and clear interface that can easily be used by scientists from diverse research domains, especially for non-computer scientists. We suspect that young-researchers (Master and PhD students) will be the major user group of the e-Science life cycle composer, while senior researcher will most likely interact with the system in terms of submitting questions and/or via the *Dataspace Browser* (see Section 3) rather than feeding the dataspace. This is actually an advantage, as it is more easier to force Master and PhD students than senior researchers to follow the e-Science life cycle model and describe

its five activities during their investigations. It is an essential task that leads to the evolution of the dataspace into a semantically rich, large-scale scientific dataspace.

Dataspace Indexer. The purpose of the Dataspace Indexer is to organize LCRs, including their subscription. It implements a storage and indexing mechanism for the scientific dataspace model described in Section 2. LCRs are organized in a flat table. Each row in the table represents a LCR. The first column stores the index of the LCR. In the other columns the indexes of individuals of the five e-Science life cycle activities are stored. The first LCR created will have index 1 in all five corresponding individuals of the e-Science life cycle activities. Starting from the second LCR the index assigned to a previous life cycle activity might occur again. This is for the purpose that an individual might be reused within another LCR. Individuals of the second LCR are therefore subscribed with either index 2 or index 1. The third row allows an index up to 3, because here again individuals are attached to either a new index 3 or in the case of re-use to the corresponding index of the re-used individual, which is in the current state of the dataspace either 1 or 2. In order to generalize this indexing mechanism let's say I_x are the indexes I of a life cycle activity $x \in \{GS, DP, TS, TE, RP\}$. Then I_x can be defined as the interval

$$[1, n] := \{i \in \mathbb{N} | 1 \leq i \leq n\}$$

Individuals of life cycle activity x are therefore indexed by elements of the interval $[1, n]$, where n is the total number of LCR available in the dataspace. Each individual corresponds to a LCR. Thus, we define

$$i_{LCR_y(x)}$$

as the index i of an individual of a life cycle activity $x \in \{GS, DP, TS, TE, RP\}$ that corresponds to the LCR y.

Dataspace Browser. The dataspace browser is a tool that allows the user to navigate through the LCRs available in the dataspace in a visual way. It sends requests to the *Query Processor*. These requests are SPARQL queries that are submitted to the RDF store. Since the response represents RDF data it can be visualized and further represented using any RDF tool available, thus the architecture allows us to integrate different tools for visualizing RDF data.

4. Implementation and Evaluation

The jSpace scientific dataspace support platform described in this paper has been implemented and evaluated in the context of two real world e-Science applications: (1) Grid-based non-invasive blood glucose measurement (NIGM) [11, 9] and (2) Grid-supported breath-gas analysis for the molecular-oriented detection of minimal diseases (BGA). The first one is based on an international research cooperation project[1]. The

[1] http://www.par.univie.ac.at/project/cadgrid/

latter one is part of the *ABA-Project* project[2]. In this section we provide some implementation details and an experimental evaluation on top of both applications.

The complete process of creating an LCR in the dataspace includes (a) the creation of semantic relationships among its datasets, (b) the indexing of each activity in the LCR, and (c) the preservation of both, the individuals and properties of the experiment into the RDF Store and its participating datasets into an appropriate data preservation system such as the integrated Rule-Oriented Data System (iRODS) [17]. This process is referred to as the *jSpace preservation process* in the following. The experimental evaluation uses an initial prototype of jSpace (v1.2), which uses the Jena framework [1] to provide persistent RDF data storage in a MySQL database and the iRODS system as to implement a basic data repository for storing dataspace participants. The performance overhead is an important factor in determining system scalability and acceptability. We evaluated jSpace for performance overhead on both, a real world NIGM and a BGA study, and discuss the results.

Experimental Setup. The following setup was used for our performance evaluation on the jSpace Preservation Process (jSpace PP) in both application scenarios.

For the execution of the BGA study, we used a desktop computer running Windows XP x64 Edition on an Intel Core 2 Quad CPU with 2.66 GHz and 4 GB of RAM. The BGA study was executed within Matlab version 7.7.0 (R2008b). The BGA study basically checks if there is some correlation in the inspiration and expiration of exhaled breath air for samples separated according by smoking behavior. It receives the data directly from within Matlab. The input dataset is represented in the Matlab data structure. The study includes three steps: (1) preparation of the data from the input dataset that is in the Matlab structure, (2) execution of the tasks, which are breath gas analytical methods implemented as Matlab functions, and (3) plotting the results and preparing a report in the HTML format.

The NIGM performance results were elaborated on a Linux server running Fedora Core 5 on an Intel Pentium D 930 CPU with 3 GHz and 4 GB of RAM. The server hosts all the services relevant for the execution of the NIGM workflow. The following software versions have been used Globus Toolkit 4.05 [18], WEEP Version 1.2.1[13], OGSA-DAI WSRF 2.2 [3] exposing a MySQL 4.0 database with the meridian measurement test datasets collected with the meridian measurement instrument. A detailed performance evaluation on the NIGM application was conducted in [4], on top of which we build our jSpace performance evaluation for that application. The NIGM service consists of the following algorithms, deployed as WS-I and WSRF-compliant CADGrid services: (1) System Identification, (2) Kalman Filtering, (3) Fast Fourier Transformation, (4) Combination Service, and finally (5) Back Propagation Neural Network.

The performance of the jSpace PP was tested on a macbook pro running Mac OS X version 10.7.1 on an 2.4 GHz Intel Core 2 Duo processor with 5 GB of RAM.

[2] http://aba.cloudminer.org/

Table 2
Performance overhead of the jSpace PP in the NIGM application

Description	Best Time	Worst Time	Avg. Overhead (time)	% Overhead
With jSpace PP	159380 ms	162425 ms	1623 ms	1.64%
Without jSpace PP	156800 ms	158453 ms	—	—

The installed software components include iRODS 2.4.1 [17], Jena 2.6.4 [1], and the current version of the jSpace API which is 1.2.

jSpace Performance Evaluation on NIGM. For the performance overhead of the jSpace PP in the NIGM application. The jSpace PP was conducted subsequent to the NIGM workflow. A comparison of the results of the execution was performed. The results of the performance overhead are shown in Table 2, which reflects that jSpace has on average 1.64% performance overhead. *Best Time* and *Worst Time* in Table 2 are the total execution times including the execution time of the NIGM-workflow. In the worst case a performance overhead of 2660ms was recorded and in the best case the overhead was 1301ms.

Figure 3 shows how the performance overhead of the jSpace PP is consolidated into its three phases. The creation of semantic relationships takes the greatest part, almost 50% of the total time needed to preserve an NIGM study, directly followed by the preservation phase of approx. 40%. The indexing phase accounts for approx. 10%.

In some application scenarios, it might be not so important to replicate the input dataset because the end point references defined by jSpace might be sufficient. However, in this experiment we have chosen to replicate the data from the dataspace participants layer, which include the input dataset as well as the result dataset and the NIGM workflow. The performance overhead of the preservation phase is mainly caused by replicating the input dataset into iRODS. The input dataset used in this experiment had the typical size of 10 MB whereas all other participating datasets were in all not larger that 3 MB on average.

jSpace Performance Evaluation on BGA. For the second application domain we have executed a typical breath gas analysis experiment with the jSpace PP subsequently to the study execution and without it. The study has been executed one hundred times in order to elaborate significant average performance results. The performance overhead is shown in Table 3, which reflects a performance overhead of approx. 11.63% on average.

We argue that the performance results presented in Table 3 represent a remarkable low performance overhead, which is most likely due to the relatively small sizes of dataspace participants in this application. Therefore, the preservation phase of the jSpace PP is quite small. Dealing with larger datasets, we are facing a great increase in the performance overhead which is mainly the result of a longer preservation phase time.

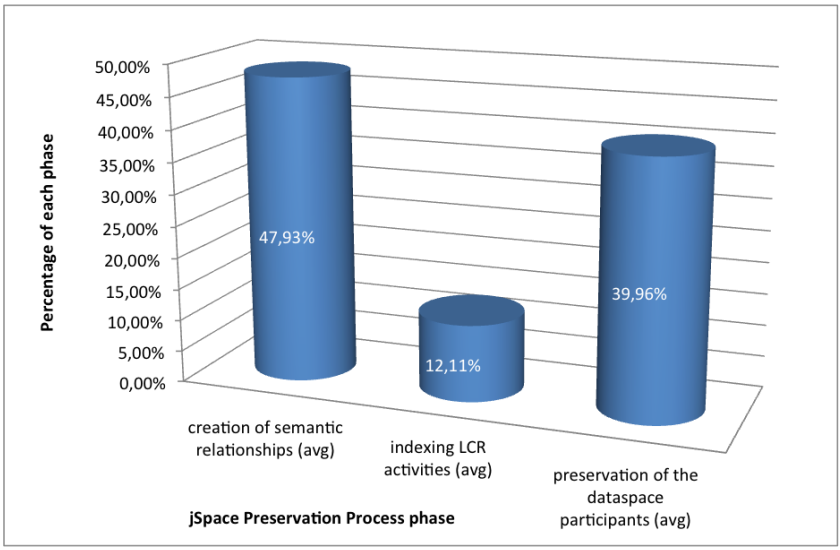

Figure 3. Distribution of the performance overhead among the phases of the jSpace PP (average on preservation of 100 NIGM-LCRs)

Table 3
Performance overhead of jSpace in the BGA application.

Description	Best Time	Worst Time	Avg. Overhead (time)	% Overhead
With jSpace	9855 ms	12078 ms	1275 ms	12.14%
Without jSpace	7915 ms	8001 ms	—	—

5. Conclusion

This paper presents a sketch of a scientific dataspace paradigm build on top of the e-Science life cycle ontology. Firstly, a theoretical model is described and based on it we propose a system architecture for a scientific dataspace support platform. Key to the platform is a semantic model for the creation, representation, and advanced searching of relationships among participants of a scientific dataspace. A prototype has been implemented and evaluated in the context of two real world e-Science applications.

Acknowledgements

The research presented in this paper was partially funded by the Austrian Federal Ministry for Transport, Innovation and Technology and the Austrian Science Fund. The Faculty of Computer Science at the University of Vienna supported this work.

References

[1] Jena — A Semantic Web Framework for Java. Project Website, 2011. http://jena.sourceforge.net/.

[2] Halevy A., Franklin M., Maier D.: *Principles of dataspace systems.* PODS, 2006.

[3] Antonioletti M., Hong N. P. C., Hume A. C., Jackson M., Karasavvas K., Krause A., Schopf J. M., Atkinson M. P., Dobrzelecki B., Illingworth M., McDonnell N., Parsons M., Theocharopoulous E.: *Ogsa-dai 3.0 – the what's and whys.* In UK e-Science All Hands Meeting, 2007.

[4] Brezany P., Elsayed I., Han Y., Janciak I., Wöhrer A., Novakova L., Stepankova O., Zakova M., Han J., Liu T.: *Inside the nigm grid service: Implementation, evaluation and extension.* Proc. of the 2008 Fourth International Conference on Semantics, Knowledge and Grid, pp. 314–321, Washington, DC, USA, 2008. IEEE Computer Society.

[5] Das Sarma A., Dong X., Halevy A.: *Bootstrapping pay-as-you-go data integration systems.* SIGMOD, pp. 861–874, 2008.

[6] Dittrich J.-P.: *iMeMex: A platform for personal dataspace management.* Proc. of SIGIR Workshop on Personal Information Management (PIM), 2006.

[7] Dong X., Halevy A.: *Indexing dataspaces.* SIGMOD, pp. 43–54, 2007.

[8] Elsayed I., Ludescher T., Schwarz K., Amann A., Feilhauer T., Brezany P.: *Towards realization of scientific dataspaces for the breath gas analysis scientific community.* Austrian Grid Draft Report, 2009.

[9] Elsayed I., Muslimovic A., Brezany P.: *Intelligent dataspaces for e-science.* CIMMACS '08, 2008.

[10] Elsayed I., Brezany P.: *Towards large-scale scientific dataspaces for e-science applications.* Proc. of the 15th international conference on Database systems for advanced applications, DASFAA'10, pp. 69–80, Berlin, Heidelberg, 2010. Springer-Verlag.

[11] Elsayed I., Han J., Liu T., Wöhrer A., Khan F. A., Brezany P.: *Grid-enabled non-invasive blood glucose measurement.* ICCS, pages 76–85, 2008.

[12] Franklin M., Halevy A., Maier D.: *From databases to dataspaces: A new abstraction for information management.* SIGMOD, 2005.

[13] Janciak I., Kloner C., Brezany P.: *Workflow enactment engine for wsrf-compliant services orchestration.* Proc. of the 2008 9th IEEE/ACM International Conference on Grid Computing, GRID '08, pp. 1–8, Washington, DC, USA, 2008. IEEE Computer Society.

[14] Jeffery S. R., Franklin M. J., Halevy A. Y.: *Pay-as-you-go user feedback for dataspace systems.* SIGMOD, pp. 847–860, 2008.

[15] Li Y., Meng X.: *Research on personal dataspace management.* IDAR, pp. 7–12, 2008.

[16] Prud'hommeaux E., Seaborne A.:. *SPARQL query language for RDF.* http://www.w3.org/TR/rdf-sparql-query/, 2008.

[17] Rajasekar A., Wan M., Moore R., Schroeder W.: *A prototype rule-based distributed data management system.* HPDC workshop on "Next Generation Distributed Data Management, 2006.

[18] The Globus Alliance: *Globus toolkit.* Website, May 2011. `http://www.globus.org/toolkit/`.

[19] W3C: *Resource description framework (RDF).* `http://www.w3.org/RDF/`, 2003.

[20] W3C: *Web ontology language (OWL).* `http://www.w3.org/2004/OWL/`, 2004.

Affiliations

Ibrahim Elsayed
University of Vienna, Research Group Scientific Computing, Nordbergstrasse 15/C/3,
`elsayed@par.univie.ac.at`

Peter Brezany
University of Vienna, Research Group Scientific Computing, Nordbergstrasse 15/C/3,
`brezany@par.univie.ac.at`

Received: 12.12.2011
Revised: 29.01.2012
Accepted: 30.01.2012